Weakly-supervised approaches for sound event detection

*Thomas Pellegrini*
*colab. Léo Cances, Patrice Guyot*

16-04-2019
Outline

Bird’s eye view on Recurrent Neural Networks

Weakly-labeled SED
- Motivation
- Problem statement
- Baseline approach
- Attention models
- Multiple Instance Learning approaches
ANR JCJC Project (ANR-18-CE23-0005-01)

- Lightly-supervised and Unsupervised Discovery of Audio Units using Deep Learning
- Start: 01 Oct. 2018
- End: 31 March 2022 (duration: 42 months)
- PhD student: Léo Cances, Post-doc to be hired
Outline

Bird’s eye view on Recurrent Neural Networks

Weakly-labeled SED
  Motivation
  Problem statement
  Baseline approach
  Attention models
  Multiple Instance Learning approaches
Generic RNNs

- A unrolled RNN

**RNN:**

- a hidden state $h$
- an optional output $y$ which operates on a variable-length sequence $x = (x_1, \ldots, x_T)$

At each time step $t$, $h_t$ is updated by:

$$h_t = f(h_{t-1}, x_t)$$

where $f$ is a non-linear function such as a sigmoid or an LSTM, or a GRU, etc.
Simple RNN (keras version)

\[ h_t = x_t \cdot W \]
\[ y_t = \tanh(h_t + y_{t-1} \cdot U) \]

Example: 10 cells, \( x_t \in \mathbb{R}^d \rightarrow W \in \mathbb{R}^{d \times 10}, U \in \mathbb{R}^{10 \times 10}, h_t, y_t \in \mathbb{R}^{10} \)

Unrolled RNN

\[
\begin{align*}
&h_0 \\
&x_0 \\
\end{align*}
= \begin{align*}
&A \\
&x_0 \\
\end{align*}
= \begin{align*}
&A \\
&x_1 \\
\end{align*}
= \begin{align*}
&A \\
&x_2 \\
\end{align*}
= \begin{align*}
&A \\
&x_t \\
\end{align*}
= \begin{align*}
&A \\
&x_t \\
\end{align*}
= \begin{align*}
&A \\
&x_t \\
\end{align*}
Long Short Term Memory (LSTM)

[Hochreiter and Schmidhuber, 1997], [Gers, 2000]
Image from https://www.deeplearningbook.org/contents/rnn.html
Gated Recurrent Units (GRU)

- Two gating units instead of three in LSTM
- shown to exhibit better performance on smaller datasets

[Cho et al, 2014], [Chung et al, 2014]
Gated Recurrent Units (GRU)

- Hidden-state update candidate at time $t$:
  \[ \tilde{h} = \tanh(W_h[\Gamma_r \odot h^{t-1}, x^t] + b_h) \]
- Actual hidden-state output:
  \[ h^t = \Gamma_z \odot \tilde{h} + (1 - \Gamma_z) \odot h^{t-1} \]
- Gates:
  - Reset gate:
    \[ \Gamma_r = \sigma(W_r[h^{t-1}, x^t] + b_r) \]
  - Update gate:
    \[ \Gamma_z = \sigma(W_z[h^{t-1}, x^t] + b_z) \]

Figure 2: An illustration of the proposed hidden activation function. The update gate $z$ selects whether the hidden state is to be updated with a new hidden state $\tilde{h}$. The reset gate $r$ decides whether the previous hidden state is ignored. See
An RNN can be trained to predict the next symbol in a sequence.

In that case, the output at each timestep $t$ is the conditional distribution

$$p(x_t | x_{t-1}, \ldots, x_1)$$

For language modeling for instance, we would use a softmax function:

$$p(x_{t,j} | x_{t-1}, \ldots, x_1) = \frac{\exp(w_j h_t)}{\sum_{i=1}^K \exp(w_i h_t)}$$

for all possible words $j = 1 \ldots K$, where $w_j$ are the rows of a weight matrix $W$.

By combining these probabilities, we can compute the probability of the sequence $x$ using

$$p(x) = \prod_{t=1}^T p(x_t | x_{t-1}, \ldots, x_1)$$
Summary of RNN types

One to one

One to many

Many to one

Many to many

Many to many

Andrew Ng
Outline

Bird’s eye view on Recurrent Neural Networks

Weakly-labeled SED
  Motivation
  Problem statement
  Baseline approach
  Attention models
  Multiple Instance Learning approaches
Machine listening

... extracting information from sound
Machine listening

... making sense of sound
Why weakly-supervised learning?

https://github.com/CrowdCurio/audio-annotator/

Why weakly-supervised learning?

Issues regarding data annotation

- What to annotate? Which granularity?
- Annotations are limited, expensive and time-consuming
- Annotations may introduce biases
- Annotations are noisy, ”label-noise”
- ”Digital labor”, les travailleurs du clic, Antonio Casilli
Sound Event Detection

- Multi-label classification problem
- Weak labels: audio tags, $y \in \{0, 1\}^C$ $\rightarrow$ many-to-one RNN
- Strong labels: frame-level tags, $Y \in \{0, 1\}^{T \times C}$ $\rightarrow$ many-to-many RNN
Weakly-labeled SED: problem statement

- We want to perform:
  - "audio tagging": infer weak labels, \( \hat{y} \)
  - "localization" or SED: infer strong labels, \( \hat{Y} \)

- Input: a sequence of T.F. representation vectors,
  \( X = \{x_1, \ldots, x_T\} \)

- Only weak labels are available for learning

- In a recording weakly labeled as \textit{Dog}, some acoustic frames will comprise dog barking and others will not
DCASE 2018 task 4 dataset

- A subset of Audioset
- 10-second audio clips
- 10 classes
- Labeled in-domain data
- Train: 4.4 h, 1578 files, 2244 occurrences, 20% for dev
- Evaluation: 0.8 h, 288 files with strong labels
Weakly-labeled SED: approaches

Three main approaches:

1. Baseline brute-force approach: "False strong labeling"
2. Attention mechanisms: Gated Linear Units, for ex.
3. MIL-inspired approach: max MIL pooling function and variants
Models: Convolutional Recurrent Neural Networks

Tags, 1x10

Global Average or Max

sigmoid

time-distr., dense, 10

time distr., dense, 64

forw. GRU backw. GRU

SED, 431x10

(spatial) dropout, p=0.2

max-pooling, 1x4

ReLU

batch-normalization

conv, 3x3, 64

log Mel filterbank coefficients: 64 bands - 431 frames
Loss function for classification: cross-entropy

"False strong labeling" (FSL): strong labels == weak label

\[
\text{loss}(\{X^k, y_c^k\}) = \text{binCE}(y_c^k, \hat{y}_{tc}^k)
\]

\[
\text{binCE}(y_c^k, \hat{y}_{tc}^k) = -y_c^k \log \hat{y}_{tc}^k - (1 - y_c^k) \log (1 - \hat{y}_{tc}^k)
\]
Attention-based approaches

1. Weighted-Gated Recurrent Units (w-GRU)
2. Gated Linear Units (GLU)
3. Other attention mechanisms
Weighted-GRU

At training time:

\[ h_t = g(Wx_t + Uh_{t-1} + b) \]

At inference time:

\[ h_t = g(Wx_t + U\omega h_{t-1} + b) \]
Weighted-GRU

Limitations: ad-hoc approach, does not work with all sound types

SOUND EVENT DETECTION FROM WEAK ANNOTATIONS: WEIGHTED GRU VERSUS MULTI-INSTANCE LEARNING

Léo Cances, Thomas Pellegrini, Patrice Guyot
Gated Linear Units (GLU)

- Pre-dominant approach to language modeling: recurrent neural networks
- GLU: non-recurrent alternative with stacked convolutions, allow parallelization over sequential tokens
Gated Linear Units (GLU)

$h_l(X) = (X \ast W + b) \odot \sigma(X \ast V + c)$

- $X \in \mathbb{R}^{N \times m}$ input of layer $l$
- learnable: $W, V \in \mathbb{R}^{k \times m \times n}, b, c \in \mathbb{R}^n$
- Similar to LSTM, the $\sigma(X \ast V + c)$ gates control the information pass
- The gates replace standard activation functions (such as ReLU)
- Linear activation to help gradients flow
- Input shift to prevent using information from future words
- GLUs wrapped in a residual block
- Output activation function: adaptive Softmax [Grave et al, 2016]
Gated CNN in SED

LARGE-SCALE WEAKLY SUPERVISED AUDIO CLASSIFICATION USING GATED CONVOLUTIONAL NEURAL NETWORK

Yong Xu*, Qiuqiang Kong*, Wenwu Wang, Mark D. Plumbley

Diagram of the proposed model architecture with gated CNN blocks and Bi-RNN for localization and weighted average.
Gated CNN in SED

Final audio tag prediction:

\[ y_c = \frac{\sum_t^T z_c^{\text{cla}}(t) \odot z_c^{\text{att}}(t)}{\sum_t^T z_c^{\text{att}}(t)} \]
Multiple Instance Learning

- MIL terminology
  - $X = \{x_1, \ldots, x_T\}$: a bag or a set
  - $x_i$: an instance
  - No dependency nor ordering among instances
  - A single binary label $Y$
  - Unknown individual labels: $y = \{y_1, \ldots, y_T\}$

MIL assumptions:

$$Y = \begin{cases} 
0, & \text{iff } \sum_t y_t = 0, \\
1, & \text{otherwise}
\end{cases}$$

In a compact form:

$$Y = \max_t \{y_t\}$$
MIL: how to model the bag-level prob $P(Y = 1|X)$?

- MIL assumption: $P(Y = 1|X)$ must be permutation invariant since no ordering nor dependency between instances $x_t$
- Instances: "monomials"

**Theorem 1.** A scoring function for a set of instances $X$, $S(X) \in \mathbb{R}$, is a symmetric function (i.e., permutation-invariant to the elements in $X$), if and only if it can be decomposed in the following form:

$$S(X) = g \left( \sum_{x \in X} f(x) \right), \quad (3)$$

where $f$ and $g$ are suitable transformations.

**Theorem 2.** For any $\varepsilon > 0$, a Hausdorff continuous symmetric function $S(X) \in \mathbb{R}$ can be arbitrarily approximated by a function in the form $g \left( \max_{x \in X} f(x) \right)$, where max is the element-wise vector maximum operator and $f$ and $g$ are continuous functions, that is:

$$|S(X) - g \left( \max_{x \in X} f(x) \right)| < \varepsilon. \quad (4)$$
MIL: how to model $P(Y = 1|X)$?

Instance-level approach

- $f$: instance-level classifier that returns scores for each instance
- $\sigma$: aggregates the scores
- $g$: Identity function

Bag-level (embedding-level) approach

- $f$: maps instances to embeddings
- $\sigma$: maps the emb. to a single bag emb.
- $g$: bag-level classifier
- $\sigma$: ”MIL pooling”, basic non-learnable choices:
  - $z = \max_{t=1 \ldots T}\{h_t\}$
  - $z = 1/T \sum_{t=1}^{T}\{h_t\}$
MIL loss for WL-SED

\[
\text{loss}(\{X, y_c\}) = \text{binCE}(y_c, \max_t \hat{y}_{tc})
\]  \hspace{1cm} (1)

where

- \( y_c \in \{0, 1\} \): ground truth label for class c
- \( \hat{y}_{tc} \in [0, 1]^T \): temporal predictions for class c
System overview

- **Audio Tagging**
  - CNN
  - Bin CE loss
  - Scores: $\mathbb{R}^{T \times C}$

- **Localization**
  - CRNN
  - MIL loss + penalty

- **Output**
  - $y \in \{0, 1\}^C$
  - Rescaling
  - Smoothing
  - Threshold
  - 0.00 2.34 speech
  - 3.19 4.52 dog
Example KO

- Issue in distinguishing some classes
Confusions between *Dog* and *Cat*, why not
Confusions between *Dishes* and *Frying*??
Prediction correlations

- Confusions between *Dog* and *Cat*, why not
- Confusions between *Dishes* and *Frying*??

→ A caveat of the MIL pooling function (here max)
  - When a class most often co-occurs with another one
  - When max predictions are scarce (a single frame possibly): during training the gradient flows on those frames only
Prediction correlations

- Confusions between *Dog* and *Cat*, why not
- Confusions between *Dishes* and *Frying*?

→ A caveat of the MIL pooling function (here max)
  - When a class most often co-occurs with another one
  - When max predictions are scarce (a single frame possibly): during training the gradient flows on those frames only

Solutions?

- Choose another MIL pooling function (ongoing work)
- Penalize for prediction similarity (joint work with Léo Cances, arxiv.org/abs/1901.03146)
Adding a similarity penalty

\[
\text{loss} (\{X, y_c\}) = \text{binCE}(y_c, \max_t \hat{y}_{tc}) \\
+ \alpha y_c \sum_{l \neq c} \max(0, \cos(\hat{y}_l, \hat{y}_c))
\]
## Results on Eval 2018

<table>
<thead>
<tr>
<th>Approach</th>
<th>Official Baseline</th>
<th>MIL</th>
<th>GLU-MIL</th>
<th>GLU-MIL+cos</th>
<th>JiaKai</th>
<th>Liu</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-score (%)</td>
<td>10.8</td>
<td>18.86</td>
<td>22.60</td>
<td>26.20</td>
<td>32.4</td>
<td>29.9</td>
</tr>
<tr>
<td>Alarm / bell / ringing</td>
<td>4.8</td>
<td>30.1</td>
<td>29.0</td>
<td>30.4</td>
<td>49.9</td>
<td>46.0</td>
</tr>
<tr>
<td>Blender</td>
<td>12.7</td>
<td>28.8</td>
<td>23.3</td>
<td>27.7</td>
<td>38.2</td>
<td>27.1</td>
</tr>
<tr>
<td>Cat</td>
<td>2.9</td>
<td>22.8</td>
<td>28.7</td>
<td>30.3</td>
<td>3.6</td>
<td>20.3</td>
</tr>
<tr>
<td>Dishes</td>
<td>0.4</td>
<td>1.0</td>
<td>0.0</td>
<td>19.0</td>
<td>3.2</td>
<td>13.0</td>
</tr>
<tr>
<td>Dog</td>
<td>2.4</td>
<td>20.1</td>
<td>19.8</td>
<td>20.9</td>
<td>18.1</td>
<td>26.5</td>
</tr>
<tr>
<td>Electric shaver / toothbrush</td>
<td>20.0</td>
<td>7.7</td>
<td>6.2</td>
<td>19.1</td>
<td>48.7</td>
<td>37.6</td>
</tr>
<tr>
<td>Frying</td>
<td>24.5</td>
<td>0.0</td>
<td>27.6</td>
<td>21.2</td>
<td>35.4</td>
<td>10.9</td>
</tr>
<tr>
<td>Running water</td>
<td>10.1</td>
<td>17.9</td>
<td>13.3</td>
<td>13.2</td>
<td>31.2</td>
<td>23.9</td>
</tr>
<tr>
<td>Speech</td>
<td>0.1</td>
<td>36.7</td>
<td>37.6</td>
<td>35.0</td>
<td>46.8</td>
<td>43.1</td>
</tr>
<tr>
<td>Vacuum cleaner</td>
<td>30.2</td>
<td>23.5</td>
<td>40.6</td>
<td>45.2</td>
<td>48.3</td>
<td>50.0</td>
</tr>
</tbody>
</table>
Adding a similarity penalty
Prediction correlations

Less confusions!
Discussion on the cos penalty

Pros

▶ Increased the class-discriminative power of the networks

Open questions

▶ Larger variance
▶ Tends to force events to not overlap (but OK for the challenge)
▶ Better to penalize the prediction distributions? e.g. a Kullback-Leibler penalty?
▶ Worked for SED, generalization to other MIL problems?
Task 4 focus:

- do we really need real but partially and weakly annotated data or is using simulated data sufficient? or do we need both?